運用保守要件

# 信頼性

## フォールトトレランス

各コンポーネントは多重化し、システムは24時間365日、サービスを提供できること。

但し、システムメンテナンスの際のシステム停止は土日に実施する場合がある。

## 成熟度

ハードウェア・ネットワークについてはお客様にて管理となるため、MTBF（平均故障間隔）については、本書では定義しない。

## 耐用寿命

システムの使用期間は最低10年とする。

技術の進歩によりシステムが著しく陳腐化しないよう十分に配慮する。

# 保守性

## 分析のしやすさ

運用者に対して障害監視機能を提供する。

アプリケーションサーバーやDBMSの障害エラー、プログラムの例外を稼働ログに残し、障害の内容や箇所が特定できるようにする。

## 変更のしやすさ

技術者を確保しやすい言語としてC#.Netで構築する。

## 安定性

コンポーネントベースで実現され、コンポーネントの修正が他のコンポーネントに及ぼす影響が最小限となるよう配慮する。

## テストのしやすさ

単体テストは.NET標準テストフレームワークまたはNunit等のテストフレームワークを使用し、自動化する。

ただし、結合テスト・UIテストについては自動化は行わない。

# 可搬性

## 順応性

将来、タブレットでの使用を想定し、他デバイスへの拡張時の影響が少なくなるよう配慮する。

## インストールのしやすさ

クライアントソフトウェアは、顧客自身がインストールできるように、専用の対話型インストーラを用意する。その方式としては、ClickOnceを採用し、自動アップデートを伴う配信方法とする。現行システムのアンインストールは刷新システムのリリース後再検討する。

## 共存力

APIの稼働環境で特に並行稼働可能かを考慮する必要があるものはなし。

クライアントアプリケーションの稼働環境で特に並行稼働可能かを考慮する必要があるものはなし。

## 継続性

APIサーバについては、継続的デリバリーができるよう、セッション情報、プール情報などを適切に設計・設定すること。

また、今後の負荷増大を考え、なるべくオンデマンドでスケールできるアーキテクチャとすること。

## 置換性

現行システムへの差戻しができるよう設計すること。

# 安全対策

## データ保護対策

アプリケーションのDB情報は、日々バックアップすること。

# 運用管理

## エンドユーザーの利用時間および時間帯

利用時間帯は8:00～20:00とする。（日本時間）

バックアップ処理、バッチ処理を必要とする場合は、予めその所要時間を測定する。

## 保守窓口の対応内容

別途、協議する。

## ハードウェア・ネットワークの保守管理

ハードウェアについては、SUBARU様にて管理する。

## ソフトウェアの保守管理

ソフトウェアについて定期的に監視し、必要な場合には適切な対応を施す。

## ソフトウェアの構成管理・変更管理・リリース管理

Microsoft Team Foundation Serverにおいて管理する。各種の変更履歴はこれらに記録すること。

## マニュアル管理

下記、２種のユーザーに対するマニュアルをそれぞれ用意すること。尚、ツールの全体をマニュアルに組み込むこと。

* システム利用者向けマニュアル
* システム管理者向けマニュアル